

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Chen et al.	§	
	§	Group Art Unit: 2167
Serial No.: 10/674,974	§	
	§	Examiner: Timblin, Robert M.
Filed: September 30, 2003	§	
	§	
For: Dynamic Generation of an	§	
XML Schema for Back End	§	
Driven Data Validation	§	

35525

PATENT TRADEMARK OFFICE
CCUSTOMER NUMBER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on June 27, 2008.

A fee of \$510.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. : 09-0447. If, however, any additional fees are necessary, I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. : 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

This appeal has no related proceedings or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

The claims in the application are: 1, 3-11, 13-20, 22-29, and 32-38.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims canceled: 2, 12, 21, 30, and 31.

Claims withdrawn from consideration but not canceled: None.

Claims pending: 1, 3-11, 13-20, 22-29, and 32-38

Claims allowed: None.

Claims rejected: 1, 3-11, 13-20, 22-29, and 32-38.

Claims objected to: None.

C. CLAIMS ON APPEAL

The claims on appeal are: 1, 3-11, 13-20, 22-29, and 32-38.

STATUS OF AMENDMENTS

The amendments filed on December 20, 2007 have been entered. No amendments are pending.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to a method for validating a plurality of data in a backend driven environment ([0029], line 5), the method comprising: creating an XML Schema for a database ([0029], lines 1-2, FIG. 23, 206), wherein the XML Schema contains a plurality of rules for validating a plurality of data in the database ([0024]); copying the database to a hashtable ([0030], FIG. 2, 200); designating a query interval ([0030], lines 8-9, FIG. 3, 204); upon the occurrence of a query interval, comparing the database to the hashtable ([0031], FIG. 3, 212); determining if the database and the hashtable are identical ([0031], FIG. 3, 214); and when the database and the hashtable are not identical, creating a new XML Schema ([0031], line 2, FIG. 3, 218); wherein the creating a new XML Schema includes automatically updating the plurality of rules ([0031], [0024], FIG. 3, 218); and wherein a new XML Schema is created only when a determination is made that the database and the hashtable are not identical ([0031], lines 1-3; FIG. 3, 214).

B. CLAIM 11 - INDEPENDENT

The subject matter of claim 11 is directed to a first method for validating proposed additions to a database comprising: accessing an XML Schema stored in a web server's virtual root (FIG. 4, 160, [033], lines 7-8, FIG. 3, 220); checking the validity of a data using the XML Schema ([0024], original claim 11); submitting the data to a database (original claim 11); validating the data (original claim 11); and adding the validated data to the database (original claim 11, FIG. 4); wherein the XML Schema is created by a second method comprising: creating an XML Schema for a database ([0029], lines 1-2, FIG. 23, 206); copying the database to a hashtable ([0030], FIG. 2, 200); designating a query interval ([0030], lines 8-9, FIG. 3, 204); upon the occurrence of the query interval, comparing the database to the hashtable ([0031], FIG. 3, 212); determining if the database and the hashtable are identical ([0031], FIG. 3, 214); and when the database and the hashtable are not identical, creating a new XML Schema ([0031], line 2, FIG. 3, 218).

C. CLAIM 20 - INDEPENDENT

The subject matter of claim 20 is directed to a program product operable on a computer, the program product comprising: a computer-usable medium; wherein the computer usable medium comprises instructions contained in the program product comprising: instructions for creating an XML Schema for a

database ([0029], lines 1-2, FIG. 23, 206); wherein the XML Schema contains a plurality of rules for validating a plurality of data in the database [0024]; instructions for copying the database to a hashtable ([0030], FIG. 2, 200); instructions for designating a query interval ([0030], lines 8-9, FIG. 3, 204); upon the occurrence of the query interval, instructions for comparing the database to the hashtable ([0031], FIG. 3, 212); instructions for determining if the database and the hashtable are identical ([0031], FIG. 3, 214); and when the database and the hashtable are not identical, instructions for creating a new XML Schema ([0031], line 2, FIG. 3, 218); wherein the instructions for creating a new XML Schema cause the computer to automatically update the plurality of rules ([0031], [0024], FIG. 3, 218); and wherein the new XML Schema is created only when a determination is made that the database and the hashtable are not identical ([0031], lines 1-3; FIG. 3, 214).

D. CLAIM 29 - INDEPENDENT

The subject matter of claim 29 is directed to a first program product operable on a computer, the program product comprising: a computer-usable medium; wherein the computer usable medium comprises executable instructions contained in the program product comprising: instructions for accessing an XML Schema stored in a web server's virtual root (FIG. 4, 160, [033], lines 7-8, FIG. 3, 220); wherein the XML Schema contains a plurality of rules for validating a plurality of data in the database ([0024]); instructions for checking the validity of a data using the XML Schema (original claim 29); instructions for submitting the data to a database (original claim 29); instructions for validating the data (original claim 29); and instructions for adding the validated data to the database (original claim 11, FIG. 4); wherein the XML Schema is created by a second program product comprising: instructions for designating a query interval ([0030], lines 8-9, FIG. 3, 204); instructions for creating an XML Schema for the database ([0029], lines 1-2, FIG. 23, 206); instructions for copying the database to a hashtable upon the occurrence of the query interval ([0030], FIG. 2, 200), instructions for comparing the database to the hashtable ([0031], FIG. 3, 212); instructions for determining if the database and the hashtable are identical; ([0031], FIG. 3, 214) and when the database and the hashtable are not identical, instructions for creating a new XML Schema ([0031], line 2, FIG. 3, 218); wherein the instructions for creating a new XML Schema cause the computer to automatically update the plurality of rules; and wherein the new XML Schema is created only when a determination is made that the database and the hashtable are not identical ([0031], lines 1-3; FIG. 3, 214.)

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to be reviewed on appeal are as follows:

A. GROUND OF REJECTION 1

Whether claims 1, 3-11, 13-20, 22-29, and 32-38 were properly rejected under 35 U.S.C. 103(a) as being unpatentable over Ng et al. U. S. Patent Number 6,609,133 B2 (hereinafter Ng) in view of Srivastava et al. U.S. Patent Application Publication 2002/0120685 (hereinafter Srivastava) and further in view of Janzig et al. U.S. Patent Number 7,016,906 (hereinafter Janzig).

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 1, 3-11, 13-20, 22-29, and 32-38)

Claims 1, 3-11, 13-20, 22-29, and 32-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Neg et al. U. S. Patent Number 6,609,133 B2 (hereinafter Ng) in view of Srivastava et al. U.S. Patent Application Publication 2002/0120685 (hereinafter Srivastava) and further in view of Janzig et al. U.S. Patent Number 7,016,906 (hereinafter Janzig).

1. Claims 1, 3-11, 13-20, 22-29, and 32-38

a. Legal Standard

The Examiner rejected claims 1, 3-11, 13-20, 22-29, and 32-38 under 35 U.S.C. 103(a) as being unpatentable over United States Patent 6,609,133 (hereinafter Ng) in view of United States Patent Application Publication 2002/0120685 (hereinafter Srivastava) and also cites to United States Patent 7,016,906 (hereinafter Janzig). As such, Appellants interpret the rejection as being over Ng in view of Srivastava and Janzig. It is well settled that in order to establish a prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); also M.P.E.P. § 2143.03. Appellants submit that Ng, Srivastava, and Janzig, alone or in combination, do not teach or disclose Appellant's claims.

b. Construction of the terms "XML Schema" and "validate"

Appellants' term XML Schema is defined in the specification to mean "a computer file containing a plurality of rules for validating data." (See Specification, [0024]). Furthermore, the term "validate" is defined to mean "to check data against an XML Schema to determine if the data meets the requirements of the database records, tables, and fields." (See Specification, [0023]). "When an Appellants states the meaning that the claim terms are intended to have, the claims should be examined with that meaning, in order to achieve a complete exploration of the applicant's invention and its relation to the prior art." *In re Zletz*, 893 F.2d 319, 321 (Fed. Cir. 1989).

c. Independent Claims 1, 10, 20 and 29

Claims 1, 10, 20 and 29 are similar. The reasons that claim 1 distinguishes over the cited art is exemplary of how claims 10, 20, and 29 also distinguish over the cited art. Therefore, the following

discussion of claim 1 is directed to claims 10, 20, and 29 as well.

(1) Claim 1 distinguishes because Ng does not disclose an XML Schema

Claim 1 recites “creating an XML Schema for a database, wherein the XML Schema contains a plurality of rules for validating a plurality of data in the database.” The cited art fails to disclose these limitations because Ng and Janzig are silent to XML schemas and Ng, Janzig, and Srivastava are silent to using an XML schema to validate data in a database.

The Examiner alleges Ng discloses these limitations in the Final Office Action p.3 (citing to Ng col. 4, ll. 1-7 & 35-42; col. 5, ll. 45-50; and fig. 3). The term “XML Schema,” as defined by Appellants, and the term “schema” as used by Ng are different. Ng states that “a relational database stores data in tables having rows (records) and columns (fields)” and that “there is a logical structure imposed on the database” known as a schema. (Ng 1:60-64). Ng further states “[o]bject-relational mapping tools read database schema information and automatically generate source code from the database.” (Ng 2:9-11) Ng is directed to addressing a situation where a programmer has added customizations to a database and/or a data base administrator has changed the database schema. (Ng. 4:29-50). Because Appellants XML Schema is a computer file containing rules for validating data “to determine if the data meets the requirements of the database records, tables and fields,” Appellants’ XML Schema is not the same as the schema in Ng. By schema, Ng means the logical structure of the database, while Appellants’ XML Schema means rules for validating data before the data goes into the database. Moreover, Ng does not automatically update a schema—rather Ng takes an updated schema and then creates new source code for the database. (Ng. 4:29-50). Thus Ng’s disclosed process begins after an administrator has changed the schema (as Ng uses the term) for the database.

Therefore, Ng only teaches that Ng’s schema is the logical structure of a database (col. 1, ll. 60-64), Ng is silent to the use of an XML schema as defined by Appellants.

(2) Claim 1 distinguishes because Ng does not disclose “validating”

Ng is also silent as to “validating ... data in the database” since “validate” is defined by Appellants to mean checking “data against an XML Schema to determine if the data meets the requirements of the database records, tables, and fields.” Janzig is also silent to using an XML schema and is also silent as to “validating ... data in the database.” Srivastava teaches the use of an XML schema (para. [0010]), but Srivastava’s XML schema validates a service descriptor (id.), and is silent to “validating ... data in the database,” as set forth in the properly construed claim.

The Examiner, in his final office action response to Appellants' arguments, argued that Ng "teaches the use of a schema reflects the structure of a database (col. 2, lns 11-13) which also describes the number of classes and interrelationships of data." (Final Office Action, pg.17.) The Examiner further argued that Ng discloses an object model made of the schema or structure of the database (col. 5, lines 20-24) and that this object model "in part teaches validation as it may specify the attributes of a field." The Examiner stated that "teaching of changing of the attributes of a field to specify what type of value can be accepted essentially teaches validating data in the database..." and "Ng's schema 'validates' in that it ensures that the data stored within a database contains the correct interrelationship." (Final Office action, pg. 18). The Examiner's logic is flawed because he relies on a premise that Ng's object model "in part teaches validation as it *may* specify the attributes of a field." (emphasis added). In other words, the Examiner argues that validation is inherent in Ng's disclosure. But inherency cannot be established by probabilities or possibilities. The following excerpt from the MPEP, 2112 Section IV, explains the Examiner's requirements when arguing inherency:

The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993) (reversed rejection because inherency was based on what would result due to optimization of conditions, not what was necessarily present in the prior art); *In re Oelrich*, 666 F.2d 578, 581-82, 212 USPQ 323, 326 (CCPA 1981). "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill in the art. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (citations omitted)

The Examiner further states that "where Ng does not explicitly recite the use of an "XML" schema for validation, "Srivastava explicitly teaches this." The Examiner's basis for this statement is that "Srivastava teaches service description information (i.e. data) and validates it against a Service Descriptor Schema which *may* take the form of an XML schema" (emphasis added) before storing it in a database or registry. (Srivastava [0010]). Once again the Examiner relies on a premise that the Descriptor Schema "*may* take the form of an XML schema" and again the Examiner is arguing that Claim 1's validation is inherent in Srivastava. Appellants submit that the Examiner's logic is again flawed because the Examiner has not met the requirements set forth in MPEP, 2112 Section IV, as set forth above.

(3) Additional elements of claim 1 that distinguish over the cited art

Claim 1 also recites “when the database and the hashtable are not identical, creating a new XML Schema” and “wherein a new XML Schema is created only when a determination is made that the database and the hashtable are not identical.” The cited art fails to disclose these limitations because Janzig always creates a new schema.

In the final office action, the Examiner argued that Ng teaches comparing hash tables representing database tables (Ng, col. 8, lines 14-16) and “this step is carried out to determine the changes (e.g. a change in type, name or number of fields) in a database (col. 8, lines 27-28).” Changes are updated (i.e. a new schema is created) to reflect changes determined by comparison of the hashtable and the database. The Examiner argues that if the schema changes, a new schema is “essentially created.” (Final Office Action, p.19).

Janzig teaches creating new schemas and then matching the old schemas with the new schemas. Janzig col. 9, ll. 30-50. The Examiner appears to read Janzig’s matching onto the claim’s determining. Appellants note Janzig’s matching (step 217) is performed after Janzig’s creating a new schema (step 215). As such, Janzig’s system will always create the new schema regardless of the outcome of the matching. Hence, Janzig fails to teach when the matching fails, creating a new schema, much less the claim’s limitations of “when the database and the hashtable are not identical, creating a new XML Schema.” Similarly, Janzig also fails to teach creating a new schema only when the matching fails, much less the claim’s limitations of “wherein a new XML Schema is created only when a determination is made that the database and the hashtable are not identical.” Srivastava is not relied upon nor does it teach these limitations.

The Examiner argued in the final office action that he disagrees with Appellants’ interpretation of Janzig because “at least in figure 19, Janzig opens a database (reference 209) to determine a change” and “if there is no change, then Janzig does nothing (reference 211).” The Examiner states that “otherwise Janzig creates a new schema. The Examiner further argues, *inter alia*, that Janzig “describes the formation of a new schema when a database is changed. (Final Office Action, p.20). The examiner’s argument is incorrect because the term Schema in Janzig does not meet the definition of Appellants’ term XML Schema. Janzig’s Schema is “a description of a prior art database, and the dual schema file of Janzig contains a modifiable and unmodifiable copy of the Schema. (Janzig 4: 49-57). Moreover, Janzig checks to see if the time stamp is changed (FIG. 19, 211) to determine the modifiable version of the schema (in the dual schema file) has been modified the administrator. The new schema created is to

create a new unmodifiable schema—but the schema is not the schema defined by Appellants. Not only is the schema not the schema defined by Appellants, but the schema has already been altered by the administrator (that is why the time stamp is checked), and so the schema is not updated in response to a comparison of the database to a hashtable containing a copy of the database. In other words, in addition to Janzig’s schema not being the XML Schema of Appellants’ claim1, the schema has already been altered by the administrator, and therefore, does not disclose the timing set forth in claim 1.

Claim 1 recites “upon the occurrence of a query interval, comparing the database to the hashtable.” The cited art fails to teach these limitations because Srivastava fails to teach “comparing the database to the hashtable.” In the Final Office Action, the Examiner argues that Ng “teaches the comparison of a hash table of a database to determine changes (Ng, col. 8, line 10-28) and Srivastava teaches periodically searching for information to check for updates to a database (0068).” The Examiner submits that “an update to the service information represents a change in the database” and that as service information is being updated before being stored in a registry (Srivastava [0010]), that an update to this information is an update to a database. Appellants submit that the examiner is wrong for the reasons set forth below.

Srivastava teaches to “periodically search for updated service information from time to time.” Srivastava para. [0068]. The Examiner appears to be reading Srivastava’s teachings of searching for updated service information (Srivastava para. [0068]) as meeting the claim’s “upon the occurrence of a query interval, comparing the database to the hashtable.” Appellants respectfully disagree with such an interpretation. Specifically, “updated service information” is not an equivalent of either a database or a hashtable, inasmuch as “service information” is information about a service and is different from a database or a hashtable. Thus, Srivastava’s periodic searching, without more, does not meet the claim’s “upon the occurrence of a query interval, comparing the database to the hashtable.”

Srivastava paragraph [0069] teaches “includ[ing] fixed input and output values which may be used to test the operation of a specified service.” The Examiner appears to be reading Srivastava’s testing of the operation of a specified service onto the claim’s “comparing the database to the hashtable.” Appellants respectfully disagree with such an interpretation. Specifically, a service is different from a database or a hashtable. Thus, testing a service, without more, does not meet the claims “comparing the database to the hashtable.”

Thus, the cited fails to teach each element of the claim. Therefore, Appellants respectfully requests that the rejection be withdrawn.

d. Claims 3, 13, 22, and 32

The Examiner argues that Srivastava [0068] teaches “when the database and the hashtable are identical, resetting the query interval.” Srivastava [0068] merely states that “[t]he services engine will periodically search for updated service information from time to time at a location or locations specified in the Service Definition, which also specifies the frequency with which the update check is to be performed.” Specifying the frequency for update checks is not the same as resetting the query interval of the independent claim when the database and hashtable are identical.

e. Claims 4, 14, 23, and 33

The Examiner argues that Ng 8:42-58 teaches “deleting the hashtable and saving the database as a new hashtable.” The cited portion of Ng discloses an object-relational mapping tool updating an object model in regard to adding a new column or a new foreign key. Ng is silent as to deleting a hashtable and saving the database as a new hashtable because Ng discloses deleting entries corresponding a deleted state in a database structure, but not deleting a hashtable as the term is used in Appellants’ claim—i.e., a copy of the database. Appellants define hashtable in the Specification to mean “a computer file or memory allocation for storing a copy of a database, and in which the computer file or memory allocation is later compared to the database.” (Specification [0019]).

f. Claims 5, 15, 24, 34

The Examiner argues that Ng 4:53 – 5:5 and FIG. 1 teach “storing the new XML Schema in a web server’s virtual root.” Ng is silent as to storing the new XML Schema in a web server’s virtual root. Indeed, the advantage of storing the XML Schema in the web server’s virtual root is explained in the specification. “The XML Schema is stored in the web server’s virtual root so that outside parties can check the validity of proposed additions to the database and correct any deficiencies prior to sending the proposed addition to the database where the proposed addition is validated by the XML Schema.” (Specification [0029], lines 8-11). The examiner appears to argue that the virtual root is inherent in the Ng’s cited section (or elsewhere for that matter), and that furthermore, storing the SML Schema is inherent in Ng’s disclosure. Appellants submit that the Examiner’s logic is flawed because the Examiner has not met the requirements set forth in MPEP, 2112 Section IV, *supra*.

g. Claims 6, 16, 25, 35

The Examiner argues that Ng, 6:1-28 and 8:10-28 teach “wherein a limited number of tables

from the database are copied to the hashtable; and wherein upon the occurrence of a query interval, the database tables are compared to the tables in the hashtable.” Appellants define hashtable in the Specification to mean “a computer file or memory allocation for storing a copy of a database, and in which the computer file or memory allocation is later compared to the database.” (Specification [0019]). Ng uses a hash table within an object (see object 400 and hash table 404 with entry 406, 408) as a way to shorten search time. Ng’s hash table is not the same as Appellants’ defined term hashtable.

h. Claims 7, 17, 26, 36

The Examiner argues that Ng 5:60-67 and 8:10-28 teach “wherein a database metadata is copied to the hashtable; and wherein upon the occurrence of a query interval, the database metadata is compared to the metadata in the hashtable.” Appellants define hashtable in the Specification to mean “a computer file or memory allocation for storing a copy of a database, and in which the computer file or memory allocation is later compared to the database.” (Specification [0019]). Appellants submit claims 7, 17, 26, and 36 distinguish over the cited art for the same reasons as set forth above.

i. Claims 8, 18, 27, and 37

The Examiner argues that Srivastava paragraphs [0068] and [0447] teach “notifying a registered party of an update in the XML Schema.” Appellants’ term XML Schema is defined in the specification to mean “a computer file containing a plurality of rules for validating data.” (See Specification, [0024]). Srivastava discloses notification services but is silent as to “a registered party” and to “an update in the XML Schema.” Appellants submit that the definition of XML Schema distinguishes over the cited art.

j. Claim 9

The Examiner argues that Srivastava [0448] teaches “using a database trigger to indicate a change in the database.” Srivastava discloses “trigger services” but is silent as to “a database trigger” and using such a trigger to “indicate a change in the database.”

k. Claim 11

The Examiner states that Ng 4:1-7 and 35-42 teach “creating an XML Schema for a database. This limitation is discussed above in regard to claim 1, and distinguishes over the cited art for the reasons stated above.

CONCLUSION

As shown above, the Examiner has failed to state valid rejections against any of the claims. Therefore, Appellants request that the Board of Patent Appeals and Interferences reverse the rejections. Additionally, Appellants request that the Board direct the Examiner to allow the claims.

Respectfully submitted,

/Rudolf O. Siegesmund/

Rudolf O. Siegesmund
Reg. No. 37,720
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Appellants

CLAIMS APPENDIX

1. A method for validating a plurality of data in a backend driven environment, the method comprising:
 - creating an XML Schema for a database, wherein the XML Schema contains a plurality of rules for validating a plurality of data in the database;
 - copying the database to a hashtable;
 - designating a query interval;
 - upon the occurrence of a query interval, comparing the database to the hashtable;
 - determining if the database and the hashtable are identical; and
 - when the database and the hashtable are not identical, creating a new XML Schema;
 - wherein the creating a new XML Schema includes automatically updating the plurality of rules; and
 - wherein a new XML Schema is created only when a determination is made that the database and the hashtable are not identical.
2. Canceled
3. The method of claim 1 further comprising: when the database and the hashtable are identical, resetting the query interval and repeating the steps in claim 1.
4. The method of claim 1 further comprising: when the database and the hashtable are not identical, deleting the hashtable and saving the database as a new hashtable.
5. The method of claim 1 further comprising: when the database and the hashtable are not identical, storing the new XML Schema in a web server's virtual root.
6. The method of claim 1 wherein a limited number of tables from the database are copied to the hashtable; and wherein upon the occurrence of a query interval, the database tables are compared to the tables in the hashtable.

7. The method of claim 1 wherein a database metadata is copied to the hashtable; and wherein upon the occurrence of a query interval, the database metadata is compared to the metadata in the hashtable.
8. The method of claim 1 further comprising: notifying a registered party of an update to the XML Schema.
9. The method of claim 1 further comprising: using a database trigger to indicate a change in the database.
10. A first method for validating proposed additions to a database comprising:
 - accessing an XML Schema stored in a web server's virtual root;
 - checking the validity of a data using the XML Schema;
 - submitting the data to a database;
 - validating the data; and
 - adding the validated data to the database;wherein the XML Schema is created by a second method comprising:
 - creating an XML Schema for a database;
 - copying the database to a hashtable;
 - designating a query interval;
 - upon the occurrence of the query interval, comparing the database to the hashtable;
 - determining if the database and the hashtable are identical; and
 - when the database and the hashtable are not identical, creating a new XML Schema.
11. The first method of claim 10 further comprising: creating an XML Schema for a database.
12. Canceled
13. The first method of claim 10 wherein the second method further comprises: when the database and the hashtable are identical, resetting the query interval and repeating the steps in claim 10.

14. The method of claim 10 wherein the second method further comprises: deleting the hashtable and saving the database as a new hashtable.
15. The method of claim 10 wherein the second method further comprises: storing the new XML Schema in a web server's virtual root.
16. The first method of claim 10 wherein the second method further comprises: wherein a limited number of tables from the database are copied to the hashtable; and wherein upon the occurrence of a query interval, the database tables are compared to the tables in the hashtable.
17. The first method of claim 10 wherein the second method further comprises: wherein a database metadata is copied to the hashtable; and wherein upon the occurrence of a query interval, the database metadata is compared to the metadata in the hashtable.
18. The first method of claim 10 further comprising: notifying a registered party of an update to the XML Schema.
19. The first method of claim 10 further comprising: using a database trigger to indicate a change in the database.
20. A program product operable on a computer, the program product comprising:
 - a computer-usable medium;
 - wherein the computer usable medium comprises instructions contained in the program product comprising:
 - instructions for creating an XML Schema for a database;
 - wherein the XML Schema contains a plurality of rules for validating a plurality of data in the database;
 - instructions for copying the database to a hashtable;
 - instructions for designating a query interval;
 - upon the occurrence of the query interval, instructions for comparing the database to

the hashtable;

instructions for determining if the database and the hashtable are identical; and

when the database and the hashtable are not identical, instructions for creating a new XML Schema;

wherein the instructions for creating a new XML Schema cause the computer to automatically update the plurality of rules; and

wherein the new XML Schema is created only when a determination is made that the database and the hashtable are not identical.

21. Canceled
22. The program product of claim 20 further comprising: when the database and the hashtable are identical, instructions for resetting the query interval and repeating the steps in claim 20.
23. The program product of claim 20 further comprising: when the database and the hashtable are not identical, instructions for deleting the hashtable and saving the database as a new hashtable.
24. (Currently amended) The program product of claim 20 further comprising: when the database and the hashtable are not identical, instructions for storing the new XML Schema in a web server's virtual root.
25. (Previously presented) The program product of claim 20 wherein a limited number of tables from the database are copied to the hashtable; and wherein upon the occurrence of a query interval, the database tables are compared to the tables in the hashtable.
26. (Previously presented) The program product of claim 20 wherein a database metadata is copied to the hashtable; and wherein upon the occurrence of a query interval, the database metadata is compared to the metadata in the hashtable.
27. (Original) The program product of claim 20 further comprising: notifying a registered party of an update to the XML Schema.

28. (Original) The program product of claim 20 further comprising: instructions for using a database trigger to indicate a change in the database.

29. (Previously presented) A first program product operable on a computer, the program product comprising:

a computer-usable medium;

wherein the computer usable medium comprises executable instructions contained in the program product comprising:

instructions for accessing an XML Schema stored in a web server's virtual root;

wherein the XML Schema contains a plurality of rules for validating a plurality of data in the database;

instructions for checking the validity of a data using the XML Schema;

instructions for submitting the data to a database;

instructions for validating the data; and

instructions for adding the validated data to the database;

wherein the XML Schema is created by a second program product comprising:

instructions for designating a query interval;

instructions for creating an XML Schema for the database;

instructions for copying the database to a hashtable upon the occurrence of the query interval,

instructions for comparing the database to the hashtable;

instructions for determining if the database and the hashtable are identical;

and

when the database and the hashtable are not identical, instructions for creating a new XML Schema;

wherein the instructions for creating a new XML Schema cause the computer to automatically update the plurality of rules; and

wherein the new XML Schema is created only when a determination is made that the database and the hashtable are not identical.

30. (Canceled)

31. (Canceled)

32. (Previously presented) The first program product of claim 29 wherein the second program product further comprises: when the database and the hashtable are identical, instructions for resetting the query interval and repeating the steps in claim 29.

33. (Original) The first program product of claim 29 wherein the second program product further comprises: instructions for deleting the hashtable and saving the database as a new hashtable.

34. (Original) The first program product of claim 29 wherein the second program product further comprises: instructions for storing the new XML Schema in a web server's virtual root.

35. (Previously presented) The first program product of claim 29 wherein a limited number of tables from the database are copied to the hashtable; and wherein upon the occurrence of a query interval, the database tables are compared to the tables in the hashtable.

36. (Previously presented) The first program product of claim 29 wherein a database metadata is copied to the hashtable; and wherein upon the occurrence of a query interval, the database metadata is compared to the metadata in the hashtable.

37. (Original) The program product of claim 29 further comprising: notifying a registered party of an update to the XML Schema.

38. (Original) The first program product of claim 29 further comprising: instructions for using a database trigger to indicate a change in the database.

EVIDENCE APPENDIX

This appeal brief presents no additional evidence.

RELATED PROCEEDINGS APPENDIX

This appeal has no related proceedings.